Mobile Ad Hoc Networks

An Evaluation of Smartphone Technologies

B. Brownlee
Y. Liang
Royal Military College of Canada

DRDC CORA CR 2011-169
October 2011

Defence R&D Canada

Centre for Operational Research and Analysis

Land Capability Development Operational Research Team
Directorate of Land Concepts and Design

Mobile Ad Hoc Networks

An Evaluation of Smartphone Technologies

B. Brownlee
Y. Liang
Royal Military College of Canada

Prepared by:

Department of Mathematics and Computer Science
Royal Military College of Canada
Kingston, ON

Project Manager: Geoff Pond, LCDORT, 613-541-5010 x.2469
Contract Number: RMC Serial#2009-0308-SLA
Contract Scientific Authority: Geoff Pond, LCDORT, 613-541-5010 x.2469

The scientific or technical validity of this Contract Report is entirely the responsibility of the contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada.

Defence R&D Canada – CORA

Contract Report
DRDC CORA CR 2011-169
October 2011

Principal Author

Original signed by B. Brownlee

B. Brownlee

Approved by

Original signed by P. Archambault

P. Archambault
Section Head, Land Systems and Operations

Approved for release by

Original signed by P. Comeau

P. Comeau
DRDC CORA Chief Scientist

Abstract

A description of the various protocols which allow for the interoperation of various networked devices is outlined. The challenges associated with implementing a mobile *ad hoc* networking protocol on a smartphone are presented. Conceptually, “packet-injection” and “IP-spoofing” capabilities provide the capabilities required to implement ad-hoc networking using smartphones. Further analysis of the capabilities and relative merits of current market offerings are then presented in order to provide a road-map for a later proof of concept implementation.

Résumé

Les divers protocoles qui rendent possible l'interopérabilité de différents appareils réseautés sont décrits. Les défis posés par la mise en oeuvre d'un protocole de réseautage spécial mobile sur un téléphone intelligent sont aussi indiqués. D'un point de vue conceptuel, les capacits “d'injection de paquets” et “d'usurpation d'adresse IP” fournissent les moyens ncessaires pour le réseautage au moyen de téléphones intelligents. Une analyse plus poussée des possibilités et des mérites relatifs des produits actuellement sur le marché est présentée en vue de fournir un plan de mise en oeuvre pour une validation de principe ultérieure.

Executive summary

Mobile Ad Hoc Networks

B. Brownlee, Y. Liang; DRDC CORA CR 2011-169; Defence R&D Canada – CORA; October 2011.

Background: Smartphones are a potentially useful tool to field soldiers and officers alike. Conceptually, such devices could function as local network and communication hubs, transmitting a soldier's life-signs, position, ammunition consumption, sensor feeds, etc. Currently, smartphones operate using a network of cell-towers. However, the Army must be able to operate independently of a commercial cell-phone network (should no network be available). A potential solution to this challenge is a Mobile Ad-Hoc Network (MANET), which is essentially a "self-configuring infrastructureless network of mobile devices connected by wireless links". [1]

Principal results: The operating system of a smartphone (unique to each manufacturer) provides different levels of capability and different permissions to users and developers with respect to MANET development and use. As a baseline, no smartphone provides the ability to operate as a backbone node within a MANET. However, some operating systems do allow the use of "packet-sniffing" and "IP-spoofing" through installation of a library such as "libpcap". Devices which allow libpcap and how lipbcap's functions might be implemented in a MANET are discussed.

Significance of results: MANETs are currently of great interest within the academic community. However, the research done by most groups tends to involve the creation and analysis of various networking protocols for use in a MANET, with little to no indication as to how such protocols would be implemented in the devices constituting a MANET, as it is assumed that the implementations use either an open platform or a custom-made platform.

In the case of a MANET comprised of smartphones given to individual soldiers, the platform is neither open nor custom-made. This document provides a description of the non-trivial challenges associated with implementing a networking protocol in such an environment and compares the various smartphone platforms available in terms of their capabilities and the ease with which a networking protocol could be implemented on them. These results may be used to guide future work on implementing MANETs for use by the Canadian Forces and should guide any future research and development on MANET implementation within the CF.

Future work: Informative future work would consist of:

1. development of a novel protocol suite for a MANET,

2. simulation of the protocol suite using an appropriate network simulation tool, such as OPNET,
3. analysis of the protocol suite's performance against suitable criteria for the deployment of the protocol in CF operations, and;
4. proof of concept development on both an Android smartphone and an iPhone through the use of libpcap and by rewriting portions of the Android operating system.

Sommaire

Mobile Ad Hoc Networks

B. Brownlee, Y. Liang ; DRDC CORA CR 2011-169 ; R & D pour la défense Canada – CARO ; octobre 2011.

Introduction : Les téléphones intelligents pourraient être des outils utiles pour les soldats sur le terrain et les officiers. D'un point de vue conceptuel, ces appareils pourraient servir de plaques tournantes locales pour les réseaux et les communications. Ainsi, le téléphone d'un soldat pourrait transmettre ses signes vitaux, sa position, sa consommation de munition, ses données de capteurs, etc. Actuellement, les téléphones intelligents utilisent un réseau de stations de base cellulaires. L'Armée doit cependant pouvoir réaliser des opérations indépendamment d'un réseau cellulaire commercial (dans le cas où aucun réseau n'est disponible). Une solution possible à ce problème est l'utilisation d'un réseau spécial mobile (MANET), qui est essentiellement un "rseau autoconfiguré" d'appareils mobiles connectés par liens sans fil sans infrastructures". [1]

Résultats : Le système d'exploitation d'un téléphone intelligent (qui est propre à chaque fabricant) offre différents niveaux de capacité et différentes permissions aux utilisateurs et aux développeurs en ce qui a trait au développement à l'utilisation des MANET. Aucun téléphone intelligent ne peut d'emblée fonctionner comme noeud de dorsale d'un MANET. Certains systèmes d'exploitation permettent cependant l'utilisation de "l'injection de paquets" et de "l'usurpation d'adresse IP" par l'entremise de l'installation d'une bibliothèque comme "libpcap". Les appareils qui permettent l'utilisation de libpcap et la façon dont les fonctions de libpcap peuvent être utilisées dans un MANET sont passés en revue.

Portée : Les chercheurs universitaires portent un grand intérêt aux MANET. Cependant, la recherche réalisée par la plupart des groupes tend à porter sur la création et l'analyse de divers protocoles de réseautage destinés à être utilisés dans un MANET, sans vraiment indiquer comment ces protocoles seraient mis en oeuvre dans les appareils constituant le MANET : il est supposé que l'implémentation utilisera une plateforme ouverte ou une plateforme faite sur mesure.

Dans le cas d'un MANET composé de téléphones intelligents fournis individuellement à des soldats, la plateforme n'est ni ouverte ni faite sur mesure. Le présent document donne une description des défis de taille posés par la mise en oeuvre d'un protocole réseau dans un tel environnement et offre une comparaison des diverses plateformes de téléphone intelligent disponibles à l'égard de leurs capacités et de la facilité avec laquelle il serait possible d'y mettre en œuvre un protocole réseau. Ces résultats peuvent servir à guider des travaux ultérieurs sur la mise en oeuvre de MANET destinés aux Forces Canadiennes (FC) et devraient orienter toute activité de recherche et développement future sur l'implémentation de MANET au sein des FC.

Recherches futures : Des travaux ultérieurs utiles pourraient porter sur :

1. la mise au point d'un ensemble de protocoles novateurs pour un MANET ;
2. la simulation de l'ensemble de protocoles au moyen d'un outil de simulation de réseau appropriée, comme OPNET ;
3. l'analyse de la performance de l'ensemble de protocoles à l'égard de critères choisis en fonction du déploiement de protocoles au cours d'opérations des FC ;
4. le développement d'une implémentation de validation de principe sur un téléphone intelligent Android et sur un iPhone au moyen de libpcap et de la réécriture de certaines parties du système d'exploitation Android.

Table of contents

List of figures

This page intentionally left blank.

1 Introduction

Previous work funded by Defence Research and Development Canada (DRDC) developed an operational reporting and return application for the Apple iPhone. The purpose of that work was to demonstrate that providing a soldier with an iPhone (or equivalent smart phone) loaded with the reports and returns application could simplify and accelerate the process of creating and sending those reports when compared to the time required to handwrite them and verbally transmit them over the Combat Net Radio (CNR).

This application allows the soldier to use a variety of menus to easily determine the type of report to be generated and to quickly populate the paragraphs of that report. The report can then be sent via email or Short Message Service (SMS) text message to either a single recipient (e.g., the platoon commander) or a group of recipients simultaneously (e.g., the commanders of different companies).

The original intent was to expand the iPhone application to leverage other capabilities of most modern smart phones, such as the ability to record video, connect to other wireless devices/sensors and access various types of data, such as maps. The driving force behind that original intent is the growing recognition that the modern smart phone is essentially a communications hub for various sensors.

Expanding on this notion, it is not difficult to envisage a scenario where a cellphone or similar piece of equipment could connect wirelessly to small Unmanned Aerial Vehicles (UAVs), acoustic sensors, Global Positioning System (GPS) receivers and transceivers, Nuclear, Biological and Chemical Defence (NBCD) sensors and even sensors which monitor the vital signs of the soldier carrying the device. Many of these sensors are contained within the smart phone itself, while the remainder could be self-contained modules which are connected to the smart phone wirelessly via WiFi or Bluetooth.

As with any piece of sensor equipment, it is insufficient to state the equipment's purpose as "to collect data". The overarching goal is to ensure that the data is distributed to those who are best able to make use of it and to those who require it the most. In the case of the application mentioned above, the mechanism for distributing that data is through a cell phone network.

However, the Army must be able to operate independently of a commercial cell-phone network (should no network be available). A potential solution to this challenge is a Mobile Ad-Hoc Network (MANET), which is essentially a "self-configuring infrastructureless network of mobile devices connected by wireless links". [1]

In order to properly frame the problem and provide context for following discussion on platforms, a very rough introduction to the nature of communications networks must first be provided. A general discussion of networking [1] takes place in Section 2. The problem's context is then given by a description of two potential scenarios in order to provide a reasonable framework under which to base assumptions. These are presented and discussed in Sections 3 and 4.

Further detail on networking, in the context of the described scenarios, is presented in Section 5. Providing a working proof of concept using MANETs in a deployed environment requires the development of protocols and algorithms which comprise the message passing and routing capabilities of the MANET. Such protocols can (and should) first be simulated using existing technologies (such as the OPNET software suite). However, at some point, those protocols and algorithms need to be implemented on a consumer smart phone and tested in realistic field conditions. A comparison of the current Consumer Off The Shelf (COTS) hardware platforms available for such research is presented in Section 6, and recommendations for which platform(s) to use in building a MANET proof of concept are given in Section 7.

[1] For further information on network protocols and network programming, the reader is encouraged to reference Stevens' text "TCP/IP Illustrated, Volume 1". [2]

2 Networking Fundamentals

There are essentially two main components to a communications network:

- **Backbone**: the backbone of the network consists of various switches, routers and the virtual/physical links between those devices. The sole function of the backbone is to facilitate transporting information from communications endpoint to endpoint. The backbone does not typically generate any traffic of its own; and
- **Clients**: the clients consisting of smartphones, computers, sensors and any other device concerned with generating or retrieving information.

The physical links between devices can be fibre optic cables, coaxial cables, twisted pair cabling or portions of the Radio Frequency (RF) spectrum.

The Defence Wide Area Network (DWAN) is an example of a network for which the majority of the links are provided through either fibre optic cables or twisted-pair. The typical DWAN client is a desktop or laptop connected via ethernet cable to a wall mounted Data/Voice Outlet (DVO). On the other side of this DVO is another series of ethernet cables physically connecting the DWAN client to a network backbone device (usually a switch) in one of the building's communications closets. All of these switches connect (possibly through other switches) via fibre optic cabling to a router. This router provides connectivity, typically also via fibre optic cabling, to the internet and other portions of the DWAN in installations across Canada. The devices which form the network backbone can also use technologies such as satellite, microwave and cellular links for interconnecting backbone nodes.

The maximum allowable distance between a DWAN client and the switch to which it is physically connected is defined by the IEEE 802.3 Ethernet Standard [3] as approximately 100 meters. The maximum distances for fibre-optic cabling, satellite and cellular transmission are typically measured in tens of kilometres, or in the case of satellite, tens of thousands of kilometers.

A wireless network (i.e., WiFi) is logically identical to a “wired” network such as the DWAN. However, the connection between clients and backbone devices is formed wirelessly via Radio Frequency (RF) signals and is governed by the IEEE 802.11 WiFi Standard. [4] The maximum distance between two devices sharing a WiFi connection is currently limited to less than 200 meters.

There is a multitude of protocols which define how various clients and backbone nodes intercommunicate. Furthermore, each protocol (or group of protocols) defines an addressing scheme. Clients and network backbone nodes are required to have one or more addresses depending on the supported protocols. When a device is instructed to begin communicating with another device on the network, it broadcasts to all other network devices, requesting those devices reply with their addresses. Devices typically maintain a list of previously determined addresses for other devices so that time and bandwidth may be saved by simply sending information directly to those addresses, instead of first broadcasting a request to all devices. Depending on the address types required, such stored lists are known as Address Resolution Protocol (ARP) caches or Domain Name Services (DNS) caches.

2.1 Co-located Networks

Returning to the example of the DWAN computers present in most CF installations, there are often different networks which are also present in the same buildings. An example of this would be the TITAN network. Despite the fact that a TITAN terminal may only be a few feet away from a DWAN PC, and that both of their network cables eventually lead to the same communications closet, those cables connect to different devices which form the backbone of different networks. There is absolutely no interchange of communication between the two clients. The same situation would be true if the network cables were replaced with encrypted WiFi links to separate the WiFi backbone nodes - there is no logical link between the two networks. Despite their proximity, one cannot access their DWAN shared drive from a TITAN terminal.

These concepts will be explored further later in the report (with emphasis on the addressing protocols and how they are used to route information between devices). For now, it is sufficient to be aware of:

1. the difference between the devices which typically form the backbone of a network (largely unseen to the general user) and the devices which act as clients to that network,
2. the logical links and addresses defining which devices are part of a given network and which are not; and
3. the fact that multiple wireless networks can exist in the same space and be incapable of intercommunication.

The previous information will serve to provide a baseline of understanding for discussing the problems inherent in creating a MANET that will extend the capabilities offered by smartphone vendors through vendor *ad hoc* networking support. However, we will first briefly discuss what exactly a MANET is in Section 2.2 and what vendor supplied *ad hoc* networking entails in Section 2.3.

2.2 What exactly is a Mobile Ad Hoc Network?

A **wireless** *ad hoc* network refers to a network in which the specialized pieces of equipment forming the backbone of a typical network are simply not present. Instead, the client devices act as both client and backbone node, which is to say that in a standard network the client devices connect **to** the network, and in an *ad hoc* network the client devices **are** the network.

A **mobile** *ad hoc* network is identical to a wireless *ad hoc* network with the additional constraint that the devices are constantly on the move. This has the effect of constantly changing the topology of the network, thereby destroying the stability of the network topology and influencing the ability of those devices to route information between them; These moving devices **will** go in and out of range of the nearest device. This necessitates creating a system for allowing the network, as a loosely-coordinated whole, to allow devices to leave/join the network and somehow propagate the changes to the network topology across all the devices in the network.

Indeed, the problem of having to maintain routing and addressing coherency in a constantly-changing network-topology is the fundamental problem associated with MANET research. These, and the requirement to be capable of interoperating with the internet as a whole, are the primary constraints that a protocol designer must take into consideration when creating new MANET protocols.

2.3 Vendor Ad Hoc Networking Support

Vendor supplied *ad hoc* networking accomplishes the task of creating a MANET in a manner which is intended to be very easy to use. However, the problem with vendor supplied ad hoc networking support is one of scale: typically, each network is limited to having only one backbone node. This is sufficient to create a network whose range is a circle centred on the singular backbone node of a radius no larger than 200 metres. Essentially, vendor supplied *ad hoc* networking is provided as a means of connecting multiple WiFi-enabled devices to a smartphone. Current smartphones do not possess the capability to enable *ad hoc* networking between multiple smartphones with each phone acting as a backbone node.

A scenario in which a smartphone would be an asset is a dismounted infantry platoon operation. Each soldier could have a smartphone which would act as a WiFi-hub for various devices carried by the soldier, such as vital signs monitors, GPS transponders and Nuclear, Biological and Chemical Warfare (NBCW) detectors. This small Personal Area Network (PAN) is often (somewhat misleadingly) referred to by smartphone vendors as an *ad hoc* network, and the majority of smartphone platforms support this functionality by default. However, this PAN functionality only supports relaying information across a single hop, i.e., from a source device to a destination device with no more than one “backbone” device between them.

Figure 1 is given as an example. In this case, devices A and E activate their vendor supplied *ad hoc* networking support. In theory, these two devices should be able to create a single network capable of providing connectivity between all the devices shown. However, vendor supplied *ad hoc* networking support is only intended to provide PAN functionality, so each device creates its own network.

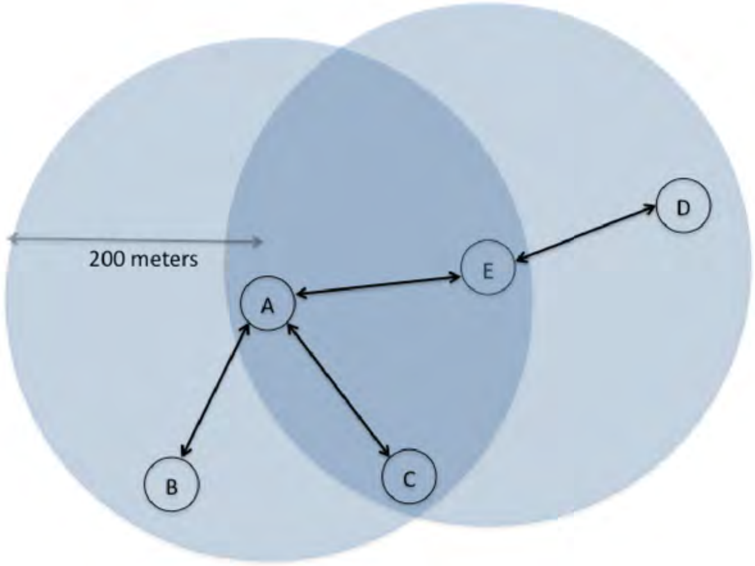

Figure 1: *A Multi-Node Vendor* Ad Hoc *Network*

In this case, devices A, B, C and E can communicate with each other through backbone node A. However, D can only communicate with E. Backbone node devices (A and E) will forward traffic between nodes, but client devices (B, C and D) will not. This means that if B attempts to send a message to E (which is out of B's WiFi range) C will receive the message and, because the message is addressed to another recipient, C will ignore it. However, A, which is a backbone node, will receive the message, verify that the recipient is on its network, and then forward the message to E. Furthermore, a device such as C will typically only be able to connect to either A's network or E's network, but not both. Additionally, the documentation is unclear regarding E's capability of having both its own network and membership in A's network.

Suppose that E is able to do both and that C has membership in A's network but not E's network. If the user of device C wishes to send a message to the user of device D (which is both out of C's WiFi range and not a member of the network C resides on), the user will be unable to do so. Devices A and E maintain a list of the devices on their respective networks and the addresses of those devices. However, those address lists are not shared. Furthermore, the addresses used are IP addresses and the backbone nodes likely choose from the same address space when assigning IP addresses to the devices in their networks. This means that a device in E's network might have the same IP address as a device in A's network. Multiple devices sharing the same IP address can result in message routing problems. However, even if such an addressing conflict does not exist, the inability of devices A and E to share routing information means that it is impossible for a device on E's network to send a message to a device on A's network, even though a logical link exists between them.

More information on addressing will be provided in Section 5. However, the crux of the problem is that vendor *ad hoc* networking support is not intelligent enough to create a single network consisting of multiple backbone nodes. Instead, it creates multiple networks each consisting of a single node, and those nodes are incapable of passing information from one network to another. The importance of this will be demonstrated in the scenarios described in Sections 3 and 4.

3 A Typical Scenario

As shown in Figure 2, a LAV III with the equipment necessary to act as a cellphone tower, is in support of a dismounted infantry unit. In this scenario, it is extremely likely that all members of the unit would be within range of the vehicle, as most cellphone technologies appear to have ranges measured in the tens of kilometres. In such a case, if soldier 11A [2] wanted to send a message (voice, email or short message service (SMS)) to soldier 119, the message would be received by the equipment in the LAV III and then forwarded to soldier 119.

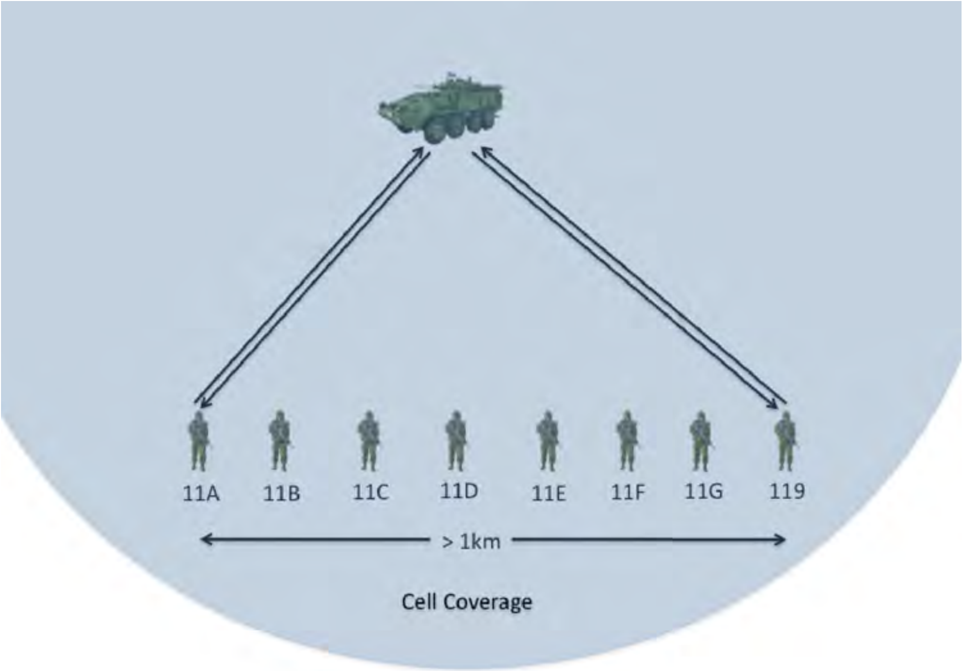

Figure 2: *The Simple Case - Vehicle With a Cellular Links*

In this case, the network has a simple hub and spoke topology. No further research is needed to develop such a network, other than building a proof of concept application which would automatically collect information from each soldier's personal network, manipulate it in some way and then forward it on to those who require the information. Current networking protocols function well in this environment and all modern smartphones being

[2] It should be noted that this method of allocating unique addresses to all of the soldiers of unit 11 (likely 1 Platoon of A Coy) does not align with the method of allocating callsigns in a standard radio network. Currently, no CF standard exists for assigning a callsign to every individual within an infantry platoon. This is an additional challenge which must be addressed before a soldier-level network like this can be deployed in order to determine where messages are being sent from.

considered for this project are capable of communicating in this environment without any modification.

In essence, this scenario is a “non-problem”, as it requires very little effort in the way of network programming to facilitate. However, it is included in this paper as a default-state. Any solution developed for providing reliable communications in a more complicated environment (such as the one described in Section 4) must be capable of operating normally in this hub and spoke environment.

As an example, suppose that each of the soldiers in Figure 2 had a PAN consisting of heart-rate monitors and a “lab on a chip” NBCW detector. It is not unrealistic to imagine a scenario where soldier 119 and 11A are separated by well over 400m. An important potential use of the technologies being discussed in this paper would allow for a near instant warning to be disseminated to all the soldiers in the diagram if, say, soldier 119’s NBCW monitor was triggered a few moments too late and his heart-rate dropped to zero at nearly the same time.

In this case, if the soldiers’ smartphones were programmed to transmit such information immediately, soldier 119’s smartphone would broadcast the warning to all devices on its network. Since the LAV III is acting as a node in the network backbone, it would receive this information, repackage it and rebroadcast it to the smartphones of soldiers 11A to 11G, letting them know that a member of their team is down, and that they are about to be subjected to an NBCW attack. This entire process would require less than a few hundred milliseconds to complete.

4 The Failover Scenario

In computing, failover refers to an automated ability to switch to a secondary or redundant system. A plausible situation requiring this capability is one where the infrastructure providing cell-tower connectivity to the devices is rendered inoperative. In the case of a deployed operation where actual cellphone towers are unavailable, the likely candidate for providing such connectivity would be a LAV III or similar vehicle. Such a situation, as discussed in Section 3, is simple (in terms of being able to provide end-to-end connectivity between devices located outside of WiFi range of one another). However, a seamless failover is required should that vehicle be rendered inoperative (e.g., destruction of the vehicle as a whole or of the antennas on the vehicle).

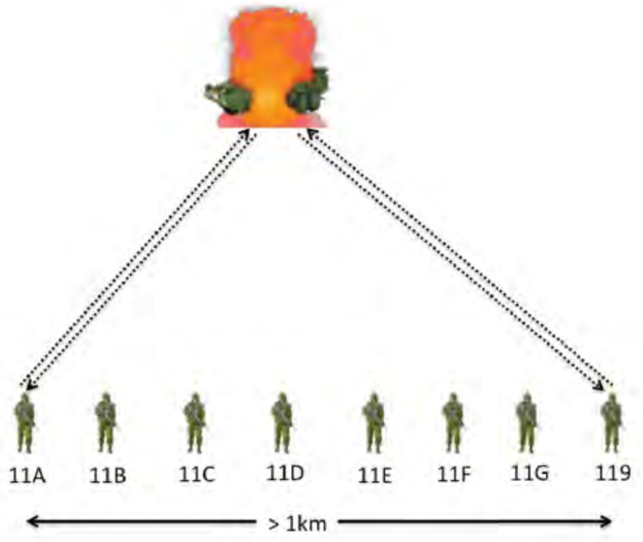

Figure 3: *The Complex Case - No Cell Tower Connectivity Available*

Suppose that each of the soldiers in Figure 2 have a PAN consisting of heart-rate monitors and a “lab on a chip” NBCW detector and that soldier 119 and 11A are separated by as much as 1000m. However, instead of a scenario consisting of a soldier quietly flat-lining because of an NBCW attack, the scenario is modified so that the NBCW attack is immediately preceded by a rocket-propelled grenade (RPG) attack on the LAV III, destroying the vehicle.

In this case, the situation is that of Figure 3. Now, when soldier 119’s smartphone attempts to contact the rest of the smartphones in the platoon, it cannot do so via cell-tower relaying. Furthermore, because the soldiers are separated by more than twice the maximum distance of WiFi (as shown in Figure 4), no single cellphone can successfully relay information between all the devices.

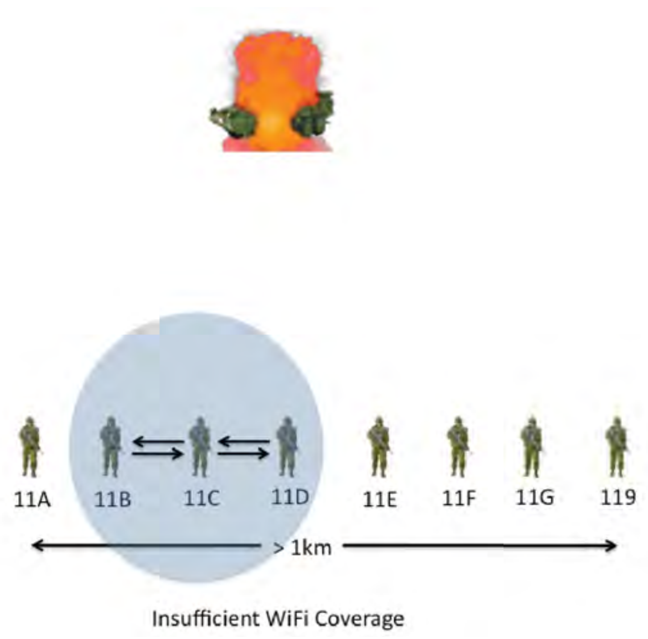

Figure 4: *The Complex Case - A Broken Cellular Link With Insufficient WiFi Coverage*

Using only the vendor-supplied version of ad-hoc networking, the messages will not be passed and the soldiers furthest from soldier 119 only receive the information regarding the NBCW attack and the death of soldier 119 if those nearest to soldier 119 are able to verbally pass that information along. However, given that soldier 119 died without making much noise, and that his death coincided with a rocket strike on the platoon's method of ingress/egress, odds are that a substantial number of valuable seconds will pass before his death is noticed. Such a failure in communication at such a critical juncture is obviously undesirable.

The ideal solution is one where the smartphones intelligently failover and act as both client and network backbone. In such a case (as shown in Figure 5), the message is relayed via WiFi from smartphone to smartphone, thus instantly alerting the platoon to the death of their comrade and the pressing need to don their Nuclear, Biological and Chemical Defence (NBCD) equipment. Unfortunately, this version of *ad hoc* networking is not supported by any of the smartphones currently available and there are challenges involved in programming existing smartphones to accomplish this task.

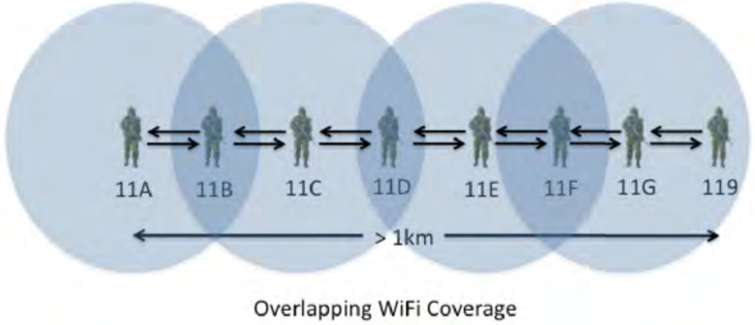

Figure 5: *The Ideal Scenario - Devices Use Overlapping Coverage to Pass Messages*

5 Networking and the Open System Interconnection Model

The International Organization for Standardization (ISO) began working on a framework for interconnecting electronic devices in 1976. The Open System Interconnection (OSI) Model [5] was the result of these efforts. Essentially, the model provides an abstract description of the tasks required for interconnecting electronic devices. These tasks are then distributed amongst seven functional layers, as shown in Figure 6.

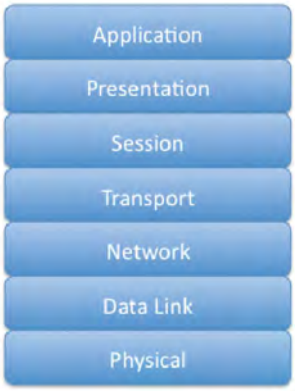

Figure 6: *The OSI Model*

All information sent over computer networks is sent in small discrete bursts. These discrete bursts of information are referred to as packets. [3] Each packet consists of three parts (some of which are optional): a header, the data and the trailer.

The Transmission Control Protocol/Internet Protocol (TCP/IP) suite is a set of communications protocols for the internet and many Local Area Networks (LANs). TCP/IP adheres to the OSI model but consolidates some of the layers. TCP/IP consists of only 4 layers, as shown in the right-hand side of Figure 7. It should be noted that the TCP/IP layers shown in Figure 7 differ slightly from the OSI layers shown in the left-hand side of Figure 7 in that the physical layer is not shown [4] and the OSI Application, Presentation and Session layers are all compressed into one TCP/IP layer: the Application layer.

This division in protocol layers tends to reflect some of the divisions and barriers presented

[3] For the remainder of this document, we will use the term packet, while recognizing that the terminology is imprecise and that other names for the discretized payloads of information (such as segment and frame) are used to differentiate between messages passed at different levels.

[4] The physical layer is typically concerned with the encoding of the binary alphabet into some sort of electrical/photonic/magnetic signal. The OSI model is concerned with specifying how that encoding/decoding is accomplished, whereas the TCP/IP protocol suite assumes that such a system is in place, but places no restrictions on how such a system should function.

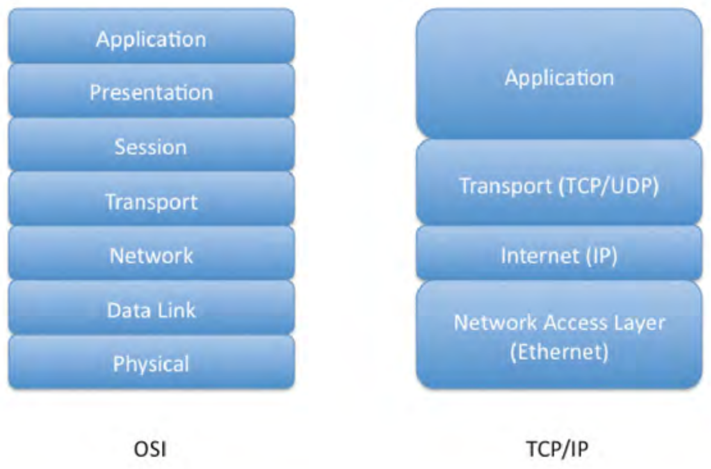

Figure 7: *Comparison of OSI and TCP/IP Layers*

to a programmer. The operating system of a client uses APIs to provide programmers with very specific types of access to the different layers. There are very few restrictions to what a programmer can access at the Application layer. However, while most conventional operating systems allow the programmer to manipulate data down at the Network Access Layer, there is no guarantee that those same permissions are made in mobile phone operating systems.

The way in which the operating system implements the TCP/IP protocol suite is known as the TCP/IP stack. Typically, when a programmer creates a message in the Application layer, the programmer uses specific APIs to pass the message to the operating system, along with information specifying the destination address. The operating system then takes that message and passes it down the layers of the stack. At each layer, headers and footers are added, allowing the message to be understood by the layers of the stack at the receiving client. These headers and footers can also be removed (and then re-added) by backbone nodes along the way, depending on the path that the message takes from client to client. Once the message reaches its destination, it is passed up the TCP/IP stack and at each layer the layer-specific headers and footers are removed.

5.1 Passing Messages

In order to understand the challenges in attempting to create a MANET, it is necessary to understand how the TCP/IP suite is used to pass messages back and forth between interconnected devices, and how those devices must behave in a multi-hop environment in order to provide that communication. The actual process of passing a message between two interconnected devices (commonly referred to as hosts) can be broken into a number of steps:

1. determine the address of the destination host,
2. determine a route to that host,
3. establish a connection [5] to the host, and
4. send and receive information.

Each device capable of connecting to a network has a Media Access Control (MAC) address. MAC addresses are unique to the device. When a device connects to a network, it is typically assigned an IP address. The backbone nodes of the network eventually form a mapping between that device's IP address and its MAC address. Programmers typically work with IP addresses, and the majority of APIs do not allow programmers to create connections between devices using MAC addresses.

When a programmer uses an API to pass a message from a source client to a destination client, the IP address of the destination client is used. The message is passed down to the operating system which adds a header and footer containing the source and destination IP address along with some other information. If the message is intended for a destination client on the same network as the source, the operating system encapsulates the message (and IP header/footer) with another header/footer containing the MAC address of the source client and the MAC address of the destination client. Then the message is transmitted (as a packet) to the backbone node that the source client is connected to. If a connection has not previously been established, the same process occurs except that the message is replaced with the first part of a three-way handshake; essentially three messages sent back and forth between the source and destination clients to ensure that both can transmit and receive to each other properly.

The backbone node, which connects the source client to the network, contains information about all the devices connected to it. It transmits this information to all the backbone nodes it is connected to. They likewise do the same, and the backbone nodes connected to them do the same, *etc*. This allows the backbone nodes to determine where the packet should be sent to next in order for it to arrive at its final destination. Once the backbone node determines the route between the source client and destination client, the messages are passed from the source client to a series of backbone nodes and then to the destination client.

These four steps are easily accomplished in a regular network. However, in an *ad hoc* network, the first two steps (addressing and routing) are more problematic.

[5] When discussing TCP/IP, the term "connection" simply refers to a link or series of links which allow for some form of logical connection between the two hosts, but not necessarily a physical connection such as a cable.

The exact methods used by the backbone nodes to compile this routing information and propagate changes is a large part of the challenge in creating a mobile *ad hoc* networking protocol suite, and are beyond the scope of this document. A discussion on the method of determining addresses follows in Section 5.2 and (while not listed above) a discussion on the problems associated with getting a client device to forward packets is given in Section 5.3.

5.2 Addressing and The Address Resolution Protocol

The Address Resolution Protocol (ARP) allows for devices to translate between unique MAC addresses and their associated IP addresses. An example of the ARP being used is presented in Figure 8.

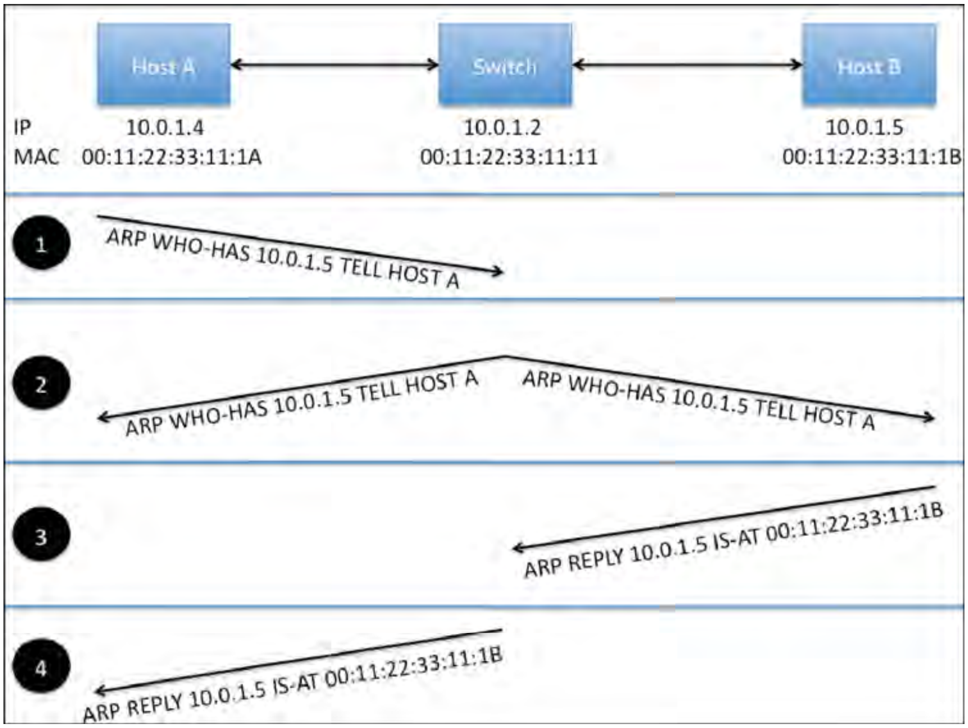
Host A
Switch
Host B
IP
MAC
10.0.1.4
00:11:22:33:11:1A
10.0.1.2
00:11:22:33:11:11
10.0.1.5
00:11:22:33:11:1B
1
ARP WHO-HAS 10.0.1.5 TELL HOST A
2
ARP WHO-HAS 10.0.1.5 TELL HOST A
ARP WHO-HAS 10.0.1.5 TELL HOST A
3
ARP REPLY 10.0.1.5 IS-AT 00:11:22:33:11:1B
4
ARP REPLY 10.0.1.5 IS-AT 00:11:22:33:11:1B

Figure 8: *The Address Resolution Protocol In Use*

The upper portion of Figure 8 shows a small network consisting of two hosts (“Host A” and “Host B”) each connected to a switch. The remainder of the figure shows the messages that are sent between the devices when Host A attempts to determine the address of Host B. This process is undertaken in order to establish a connection and begin message passing. Each step of this process is identified via a number on the left-hand side of the diagram.

Arrows travelling from left to right (or vice-versa) represent messages being passed by the devices above the arrows. Levels 1 through 4 represent the time-sequential process flow. The steps taken are as follows:

1. The first message is a request for the device assigned the IP address of 10.0.1.5 to respond to Host A. Host A sends this request out on all available links (in this case, there is only one link and it is between Host A and the switch). The switch uses the information contained in the message to build a table mapping Host A's IP and MAC address and then attempts to determine whether or not it has cached the information being requested by Host A. We assume in this case that it has not so...
2. the switch rebroadcasts the request to all devices connected to it. Host B receives the request and the operating system of Host B recognizes that the request is for the MAC address of Host B, so it generates a reply,
3. and sends that reply to the switch. The switch checks the destination address of the reply, and compares it against the cached list of addresses. The switch cached Host A's information in Step 1 so...
4. the switch forwards the message directly to Host A. Host A now updates its own cache of IP/MAC address mappings so that the next time it needs to send a message to Host B, it will not need to send an ARP WHO-HAS message requesting the information.

5.3 Forwarding Packets

When a backbone node device receives a packet, its operating system examines the packet to determine the packet's destination. Once the destination is known, the operating system determines whether or not it knows of a route to the intended destination. If a route is known, the packet is forwarded along that route. If a route is not known, either because the device has an IP address which is outside the range of IP addresses comprising the network, or because the destination device is not responding to ARP requests, the packet is silently discarded.

Conversely, when a client device receives a packet, the operating system examines it to determine its intended destination. If the packet is not destined for a MAC address or IP address in use by the operating system, the packet is silently discarded. This discarding of the packet happens before any application running on the system can access the information stored within the packet.

The only exception to this is when an operating system allows for some library (such as libpcap [6]) to place the network card into "promiscuous mode". This allows software to access the information contained within packets that would otherwise be discarded. However, putting a network card into promiscuous mode can only be done by an application which is being run with elevated user privileges.[6] Enabling a client device to listen for network traffic which is destined for other devices is known as “packet-sniffing”.

Furthermore, when a backbone node rebroadcasts the packet, the source IP and MAC addresses of the packet are kept as the addresses of the originating client, which allows the recipient to determine the address to which it should send a response. However, the default behaviour of clients is to silently drop packets which are destined for other addresses. The operating system performs this action before such packets are passed up the TCP/IP stack, and consequently, before any applications are able to access the contents of the packet. Subverting this behaviour can be accomplished by using a library such as libpcap. However, such subversion only solves half of the rebroadcasting problem; programmers can use libpcap to capture packets destined for other addresses. Programmers can interpret each packet and extract the message contained therein. However, if the programmer then uses the standard networking APIs provided by the operating system to repackage the message and send it on its way, the operating system will automatically craft headers and footers for that message. The source address field of the headers will contain the addresses of the rebroadcasting client and not those of the originating client.

The desired behaviour is for the rebroadcasting client to receive the packet, strip the message from the packet and repackage it for transmission without changing the source IP address field. In order to accomplish this, a programmer must have access to APIs which allow for the crafting of custom packets in which all fields of the packet are modifiable.[7] The creation of custom-packets is known as “packet-injection” and the creation of custom-packets whose source IP address does not match the IP address of the system generating those packets is known as “IP-spoofing”.

Client devices are required to act as backbone nodes when they form part of an *ad hoc* network. As such, they must be able to effectively forward packets between devices. Consequently, they must be able to perform packet-sniffing and packet-injection.

Furthermore, in order to have client devices properly forward addressing information (such as ARP WHO-HAS requests), those devices must be capable of performing the role of the switch as shown in Section 5.2, Figure 8. Forcing a client to behave as a switch by “faking” ARP requests and responses is known as “ARP-Spoofing” [7], and the forwarded packets must have their source IP address “spoofed” to contain the source IP of the originating system and not the rebroadcasting system.

[6] Elevated user privileges are sometimes referred to as “root privileges” in reference to the name of the super-user account on all Unix-based operating systems; the root account.

[7] These are typically referred to as raw sockets. Raw sockets differ from standard networking sockets in that the programmer can specify the exact bitfield to be transmitted across the connection. Conversely, the operating system automatically fills in certain packet fields (such as source address) when using standard sockets.

6 Comparison of Hardware Platforms

Subsequently, the Apple iPhone, Android Phones, Windows Mobile phones, and the Blackberry are all incapable of generating custom packets byte by byte unless the application attempting to do so is run with elevated privileges.[8, 9, 10, 11] This means that none of them are capable of performing packet-injection using their default APIs. However, it is very likely that the standard libraries for creating raw sockets (the mechanism used to create custom packets in conventional operating systems) will work on jailbroken phones. That being said, the use of those libraries on smartphones is unlikely to be trivial.

Platforms for which libpcap functions (as of this writing, libpcap does not function on Windows Mobile or Blackberry smartphones) will not necessarily need to be capable of creating raw sockets through operating-system-supplied APIs, as libpcap can also provide raw socket functionality.

6.1 Android

It is unclear as to whether or not libpcap can be used on an Android smartphone. In order to fulfill the requirement to perform packet-sniffing, the phone would have to allow its network card to be put into promiscuous mode, and there is some indication that doing so may not be possible.[12] However, even if it is possible, doing so will require administrator privileges on the device; i.e., it needs to be run as root. Doing so will require jailbreaking the phone, since modern smartphones do not allow users to run any software as root.

That being said, a second option for Android phones is available; it is possible to obtain source code for the Android operating system. Developers could take this (very well documented) source code, modify it and upload it onto a smartphone. Essentially this would allow a developer to modify the default behaviour of an Android phone to match the desired behaviour described in Section 4.

6.2 iPhone

An application for the iPhone named Pirni exists which uses libpcap. The Pirni application is able to put the phone's network card into promiscuous mode and can be successfully used to provide packet-sniffing and packet-injection capabilities [13], although the phone does need to be jailbroken for this to work. However, Apple is notorious for changing their APIs with major releases of their Operating System (OS) and although libpcap currently works, there is no guarantee that it will continue to work with future releases of Apple's OS.

Crafting custom packets (and performing packet-sniffing if libpcap functionality is ever disabled by an update to the operating system) may be possible through the APIs used by

Apple-employed developers when creating their bundled applications. Unfortunately, these “private APIs” (while easily found) lack formal documentation and consequently, can be very difficult to use.

Access to the iPhone operating system source code is not possible.

6.3 Windows Mobile

Windows Mobile devices are relatively new to the smartphone market. As of this writing, libpcap has not been ported to windows mobile devices. No other alternative for performing packet-sniffing and packet-injection using a Windows Mobile device is known.

6.4 Blackberry

As of this writing, no known method of performing packet-sniffing using Blackberry smartphones exists.

7 Recommendations

It is recommended that future work related to Mobile Ad Hoc Networking should consist of:

1. development of a novel protocol suite for a mobile *ad hoc* network,
2. simulation of the protocol suite using an appropriate network simulation tool such as OPNET,
3. analysis of the protocol suite's performance against suitable criteria for the deployment of the protocol in CF operations, and;
4. proof of concept development on **both** an Android smartphone and an iPhone through the use of libpcap and by rewriting portions of the Android operating system.

Until libpcap is ported for use on Blackberry or Windows Mobile, **neither platform** is recommended for proof of concept development of a mobile *ad hoc* networking protocol suite.

References

[1] Mobile ad hoc network (online), Wikimedia Foundation, Inc, http://en.wikipedia.org/wiki/Mobile_ad_hoc_network (Access Date: June 2011).

[2] Stevens, W. Richard (1993), TCP/IP illustrated (vol. 1): the protocols, Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

[3] IEEE 802.3 Ethernet Working Group (online), http://www.ieee802.org/3/ (Access Date: June 2011).

[4] IEEE 802.11 Wireless Local Area Networks (online), http://www.ieee802.org/11/ (Access Date: June 2011).

[5] Zimmermann, H. (1980), OSI Reference Model–The ISO Model of Architecture for Open Systems Interconnection, *Communications, IEEE Transactions on*, 28(4), 425 – 432.

[6] Tcpdump & LibPCAP Public Repository (online), http://www.tcpdump.org/ (Access Date: June 2011).

[7] Introduction to ARP Poison Routing (APR) (online), http://www.oxid.it/downloads/apr-intro.swf (Access Date: January 2011).

[8] iPhone Raw Socket (online), Apple, Inc., https://discussions.apple.com/thread/2171312?threadID=2171312 (Access Date: June 2011).

[9] StackOverflow - Raw Sockets on Android (online), http://stackoverflow.com/questions/2608478/raw-sockets-on-android (Access Date: June 2011).

[10] Raw Sockets (online), http://msdn.microsoft.com/en-us/library/aa922428.aspx (Access Date: June 2011).

[11] Java Development Guides and API Reference (online), Research In Motion, Inc., http://docs.blackberry.com/en/developers/subcategories/?userType=21\&category=Java+Development+Guides+and+API+Reference (Access Date: June 2011).

[12] Tcpdump Mailing Archives - Re: Libpcap on mobile Android platform (online), http://seclists.org/tcpdump/2010/q1/98 (Access Date: June 2011).

[13] Development SVN for n1mda - Pirni - Worlds first native iPhone ARP spoofer and network sniffer (online), http://code.google.com/p/n1mda-dev/ (Access Date: June 2011).

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Department of Mathematics and Computer Science
Royal Military College of Canada
Kingston, ON

2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

UNCLASSIFIED

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

Mobile Ad Hoc Networks

4. AUTHORS (Last name, followed by initials – ranks, titles, etc. not to be used.)

Brownlee, B.; Liang, Y.

5. DATE OF PUBLICATION (Month and year of publication of document.)

October 2011

6a. NO. OF PAGES (Total containing information. Include Annexes, Appendices, etc.)

38

6b. NO. OF REFS (Total cited in document.)

13

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Contract Report

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

Defence R&D Canada – CORA
Dept. of National Defence, MGen G.R. Pearkes Bldg., 101 Colonel By Drive, Ottawa, Ontario, Canada K1A 0K2

9a. PROJECT NO. (The applicable research and development project number under which the document was written. Please specify whether project or grant.)

ST000012TC01

9b. GRANT OR CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

RMC Serial#2009-0308-SLA

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DRDC CORA CR 2011-169

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

(X) Unlimited distribution
(   ) Defence departments and defence contractors; further distribution only as approved
(   ) Defence departments and Canadian defence contractors; further distribution only as approved
(   ) Government departments and agencies; further distribution only as approved
(   ) Defence departments; further distribution only as approved
(   ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11)) is possible, a wider announcement audience may be selected.)

Unlimited Distribution

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

A description of the various protocols which allow for the interoperation of various networked devices is outlined. The challenges associated with implementing a mobile *ad hoc* networking protocol on a smartphone are presented. Conceptually, “packet-injection” and “IP-spoofing” capabilities provide the capabilities required to implement ad-hoc networking using smartphones. Further analysis of the capabilities and relative merits of current market offerings are then presented in order to provide a road-map for a later proof of concept implementation.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Mobile Ad-Hoc Networking
Dismounted Soldier
Wireless Communications

DEFENCE R&D DÉFENSE

DRDC CORA

www.drdc-rddc.gc.ca